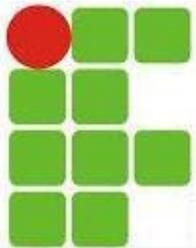
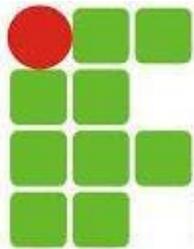


# ALGORITMOS

**Professor: Diego Oliveira**

**Aula 03 -  
Constantes, Variáveis e Memória**

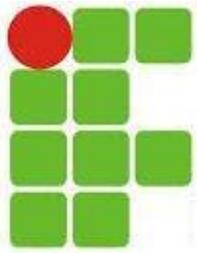




# Constantes

- Constantes são espaços de memória onde se pode armazenar um valor, de um tipo, que não será alterado ao longo do código
- Utilizam-se constantes para números conhecidos fixos, evitando assim os erros
- Uma constante é criada utilizando-se a palavra-chave 'final'
- O valor de uma constante só pode ser atribuído uma única vez!





# Constantes

- No exemplo ao lado uma constante PI é criada, após isso tenta-se alterar o valor de PI para um outro número, resultando em um erro:

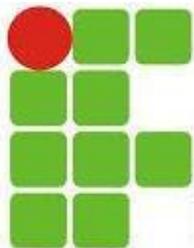
```
3  #include <stdio.h>
4
5  int main()
6  {
7      const double PI = 3.14;
8
9      PI = 3.15;|
10
11     return 0;
12 }
13
```

input

Compilation failed due to following error(s).

```
main.c: In function 'main':
main.c:9:8: error: assignment of read-only variable 'PI'
   9 |     PI = 3.15;
     |         ^
```



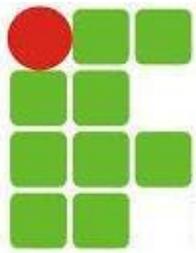


# Variáveis

- Como vimos na aula passada, as variáveis possuem um tipo e um valor e podem ter esse valor alterado a qualquer momento:

```
3  #include <stdio.h>
4
5  int main()
6  {
7      int idade = 39;
8
9      printf("Idade atual = %d \n", idade);
10
11     idade = 40;
12
13     printf("Proxima idade = %d \n", idade);
14
15     return 0;
16 }
17
```





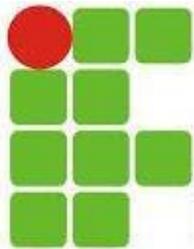
# Variáveis

- O valor da variável pode ser alterado, porém seu tipo NUNCA poderá ser trocado:

```
3  #include <stdio.h>
4
5  int main()
6  {
7      int idade = 39;
8
9      printf("Idade atual = %d \n", idade);
10
11     idade = "quarenta";
12
13     printf("Proxima idade = %s \n", idade);
14
15     return 0;
16 }
17
```

```
main.c: In function 'main':
main.c:11:11: warning: assignment to 'int' from
11 |     idade = "quarenta";
    |             ^
```



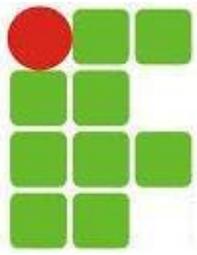


# Memória

- Cada tipo de variável ocupa um valor na memória RAM e tem um range de valores:

Tipo	Início	Fim	Memória
boolean	true	false	1 bit
short	-32768	32767	16 bits
int	-2147483648	2147483647	32 bits
long	-9223372036854775808L	9223372036854775807L	64 bits
float	1.40129846432481707e-45	3.40282346638528860e+38	32 bits
double	4.94065645841246544e-324D	1.79769313486231570e+308D	64 bits



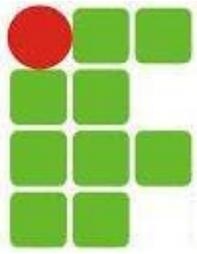


# Ler Valores do Teclado

- Para ler valores digitados pelo usuário vamos utilizar **scanf**:

```
3  #include <stdio.h>
4
5  int main()
6  {
7      int idade;
8
9      printf("Digite sua idade: ");
10     scanf("%d", &idade);
11
12     printf("Você tem %d anos.\n", idade);
13     return 0;
14
15 }
```





# Ler Valores do Teclado

- Exemplo com vários tipos:

```
int idade;
char nome[100];
double salario;
char curso;
bool isProfessor;
int prof;

printf("Digite sua idade: ");
scanf("%d", &idade);

printf("Digite seu salario: ");
scanf("%lf", &salario);

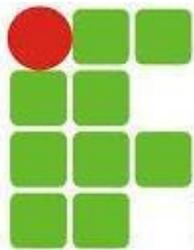
printf("É professor? ");
scanf("%d", &prof);
isProfessor = prof;

printf("Digite seu nome: ");
scanf("%s", nome);

printf("Digite seu curso: ");
scanf("%c", &curso);

return 0;
```





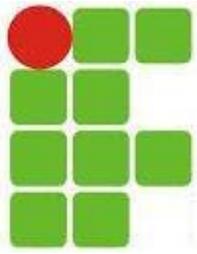
# Exercício

- Escreva um código que leia do teclado o valor do RAIO de um círculo e calcule:
  - Diâmetro
  - Perímetro
  - Área

Utilize  $\text{PI} = 3.14$  como uma CONSTANTE
- Leia também o valor da Aresta de um cubo e calcule:
  - Área da Superfície
  - Volume
  - Diagonal ( $a * \sqrt{3}$ )

Utilize  $\sqrt{3} = 1.73$  como uma CONSTANTE





# Perguntas?

